

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-161378

(43)公開日 平成11年(1999) 6月18日

(51)Int.Cl.⁹

識別記号

F I

G 0 6 F 1/20
1/08

G 0 6 F 1/00
1/04

3 6 0 D
3 2 0 Z

審査請求 有 請求項の数 4 O L (全 6 頁)

(21)出願番号

特願平9-330483

(22)出願日

平成9年(1997)12月1日

(71)出願人 000190541

新潟日本電気株式会社

新潟県柏崎市大字安田7546番地

(72)発明者 森山 秀一

新潟県柏崎市大字安田7546番地 新潟日本
電気株式会社内

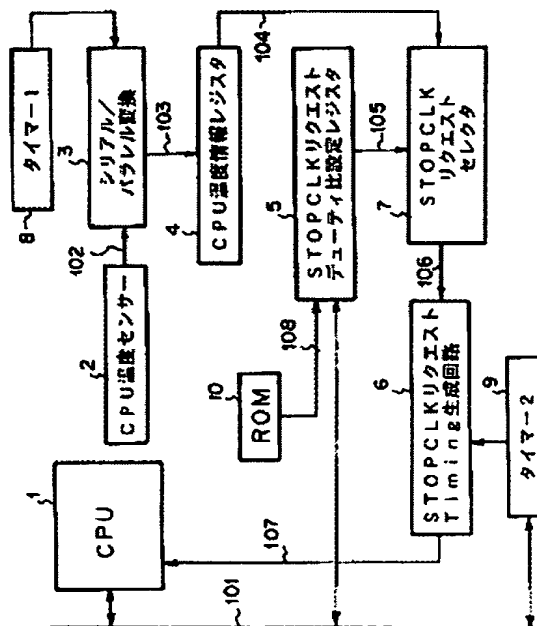
(74)代理人 弁理士 山下 稔平

(54)【発明の名称】 CPU温度制御回路

(57)【要約】

【課題】 CPU温度制御において、そのCPU温度状態での最適なCPU性能を得ること及び、CPUの温度上昇によるハードウェアの故障を防ぐ。

【解決手段】 リアルタイムにCPUの温度を検出する機能を持ち、予め設定された各CPU温度に最適にデューティ比を調節された、CPUへのSTOPCLK信号の中から、そのCPUの温度に対応した1つを選択する機能を有している。一例としては、CPU温度を検出し、その情報をリアルタイムに格納する回路と、その温度に最適なSTOPCLKリクエストを選択する回路(2, 3, 4, 7, 8)と、複数のデューティ比が異なるSTOPCLKリクエスト信号のTimingを生成する回路(5, 6, 9)と、を有することを特徴とするCPU温度制御回路である。



【特許請求の範囲】

【請求項1】 CPUの温度を検知する温度検知手段と、

CPUの温度ごとにデューティ比の異なるSTOPCLKリクエスト要求信号の発生回路と、前記検知されたCPU温度によって、CPUへのSTOPCLKリクエスト信号のデューティ比を変更してCPU性能を調節するべく、前記複数のSTOPCLKリクエストの中から最適な1つを選択してCPUへ供給するSTOPCLKリクエスト切り替え回路と、を有することを特徴とするCPU温度制御回路。

【請求項2】 CPU温度を検知し、その情報をリアルタイムに格納する回路と、その温度に最適なSTOPCLKリクエストを選択する回路(2, 3, 4, 7, 8)と、複数のデューティ比が異なるSTOPCLKリクエスト信号のTimingを生成する回路(5, 6, 9)と、を有することを特徴とするCPU温度制御回路。

【請求項3】 リアルタイムにCPUの温度を検出するCPU温度センサー2と、CPU温度センサー2からのシリアルCPU温度データ102をリアルタイムにパラレルCPU温度データ103に変換するシリアル/パラレル変換3と、リアルタイムに変化するパラレルCPU温度データ103を格納するCPU温度情報レジスタ4と、CPU温度情報レジスタ4の設定値を、一定の周期毎に更新するためのタイマー1と、CPU温度毎のSTOPCLKリクエスト信号の“H”：“L”のデューティ比データが格納されるSTOPCLKリクエストデューティ比設定レジスタ5と、CPU温度情報レジスタ4からのCPU温度デコード信号104により、そのCPU温度に一致した選択STOPCLKリクエストデューティ比データ106をSTOPCLKリクエストTiming生成回路6へ渡すSTOPCLKリクエストセクタ7と、選択したSTOPCLKリクエストデューティ比データ106とタイマー2(9)からの周期をもとにSTOPCLKリクエスト信号107を生成するSTOPCLKリクエストTiming生成回路6と、を有することを特徴とするCPU温度制御回路。

【請求項4】 前記STOPCLKリクエストデューティ比設定レジスタ5の設定値を格納したROM10を有する、ことを特徴とする請求項3記載のCPU温度制御回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、パソコン等のCPUを用いた装置に関し、特に、CPU温度制御回路に関する。

【0002】

【従来の技術】従来のCPU温度制御回路は、ソフトウェア制御によるSTOPCLKリクエストをCPUへ発生させることによりCPUの性能を調節してCPUの温度を制御していた。

【0003】また、CPU温度に応じてCPU性能を切り替えるための温度の検出ポイントは、一点しか設けることができなかった。この例としては、特開平2-207312、特開平3-62611が挙げられる。

【0004】

【発明が解決しようとする課題】従来のCPU温度制御回路は、上述したように、主に2通りの方法があるが、双方ともソフトウェアでの制御を必要とし、そのソフトウェア処理のために性能低下を巻き起こす危険性があった。

【0005】1つは、CPUの温度が、規格値より低い温度で割り込みをCPUに発生させ、ソフトウェアでの割り込み処理の中でCPUへSTOPCLKリクエストを発生させてCPUの処理能力を低下させることにより規格値を超えて高温にならないように制御していた。この方法では、CPUの温度が割り込みを発生させる温度の近辺に飽和するような場合、割り込みが多発し、その処理の回数が増加するために性能が低下するという問題があった。

【0006】もう1つの方法は、CPUの温度をソフトウェアが定期的に読みとり、温度状況によりCPUへのSTOPCLKリクエストをCPUへ発生する割合を調整する方法であったが、この方法では、常に温度の読みとりを行わなければならない、その分の性能が必ず低下していた。

【0007】すなわち、従来の技術において、第1の問題点は、ソフトウェアによる制御方法では、必ず、そのソフトウェア処理のために性能低下が発生することであり、第2の問題点は、ソフトウェアによる制御では、プログラムミス等により、ソフトウェア制御が不可能になった場合、CPUの温度の上昇を止める手段がなくなりCPUが故障する等の危険性があった。

【0008】また、特開平2-207312、特開平3-62611の方式では、ソフトウェアによる制御を必要としないが、CPU温度の検出ポイントが1点しか設けることができない為、そのCPU温度状態により、大幅な性能変動をさせなければならない、CPU温度に応じた最適な性能制御を行うことができなかった。

【0009】【発明の目的】本発明の第1の目的は、ソフトウェアの制御を必要とせず、CPUの温度状況により適切なCPU性能を制御することにより、その温度環境下でのCPU性能を十分に引き出すことである。

【0010】また、第2の目的は、CPUの各温度毎に対応したSTOPCLKリクエスト信号のデューティ比を制御することで、そのCPUの温度環境での最適な性能を供給することである。

【0011】また、第3の目的は、CPUの温度制御をソフトウェアの制御を必要としないことにより、プログラムミス等によるソフトウェアでのパソコンの制御が不能な状態に陥ってもCPUの温度を制御し続けることを可能とし、CPUの温度の上昇による故障を防ぐことである。

【0012】

【課題を解決するための手段】本発明のCPU温度制御回路は、CPUの温度状況に応じてCPUの性能を調節する回路を有している。CPUの温度が高温の場合は、CPUの性能を下げ、CPUの温度が低い場合には、CPUの性能を上げる制御をCPUへのSTOPCLKリクエスト信号の発生比率を可変することにより、CPUの性能を調節することによりCPUの温度制御を行う。

【0013】より、具体的には、CPU温度を検知、その情報をリアルタイムに格納、その温度に最適なSTOPCLKリクエストを選択する手段(図1-2, 3, 4, 7, 8)と、複数のデューティ比が異なるSTOPCLKリクエスト信号のTimingを生成する手段(図1-5, 6, 9)を有している。

【0014】

【作用】CPUの温度をCPU温度センサーが検出し、そのCPUの温度に適切なCPUの性能を、CPUへのSTOPCLKリクエストの発生比率を調節することにより行う。このためソフトウェアでCPUの温度制御を行う必要が無い。また、ソフトウェアでのパソコンの制御が不能になった場合でも、CPUの温度は制御されるため、温度上昇によりCPU等の部品が壊れることは無い。

【0015】

【発明の実施の形態】**【構成の説明】**まず発明を具体的に表わした全体構成、構成要素を図1を用いて説明する。CPU温度センサー2により、リアルタイムにCPU1の温度データをシリアル/パラレル変換3に渡す。シリアル/パラレル変換3では、CPU温度センサー2からのシリアルCPU温度データ102をリアルタイムにパラレルCPU温度データ103に変換する。

【0016】タイマー1(8)は、一定の周期タイマーで、リアルタイムに変化するパラレルCPU温度データ103をCPU温度情報レジスタ4への更新周期として使用される。CPU温度情報レジスタ4の設定値は、タイマー1(8)の周期毎に更新される。STOPCLKリクエストデューティ比設定レジスタ5には、CPU温度毎のSTOPCLKリクエスト信号の“H”：“L”のデューティ比データが格納される。

【0017】STOPCLKリクエストデューティ比設定レジスタ5の設定値は、初期設定値よりCPUアドレス/データバス101を介してCPU1による設定値の変更、あるいはROMデータバス108を通じてROM10のデータをダウンロードしてSTOPCLKリクエ

ストデューティ比設定レジスタ5の設定値を変更することも可能である。特にROM10によるダウンロード機能は、ROM10のデータをCPUの発熱状態が異なるパソコン装置毎に用意することにより、H/W及びBIOSを変更せずに使用できるという利点がある。

【0018】STOPCLKリクエストデューティ比設定レジスタ5のCPU温度毎の設定値は、STOPCLKリクエストデューティ比データ105を通じてSTOPCLKリクエスト発生比率データセクタ7に入力される。

【0019】STOPCLKリクエストセクタ7では、CPU温度情報レジスタ4からのCPU温度デコード信号104により、そのCPU温度に一致した選択STOPCLKリクエストデューティ比データ106をSTOPCLKリクエストTiming生成回路6へ渡す。STOPCLKリクエストTiming生成回路6では、選択STOPCLKリクエストデューティ比データ106とタイマー2(9)からの周期をもとにSTOPCLKリクエスト信号107を生成する。タイマー2(9)の周期は、CPUアドレス/データバス101を介してCPU1により周期設定の変更も可能である。

【0020】**【動作の説明】**本発明の動作を図1、図2、図3を用いて説明する。図1は、上記で説明した本発明の構成を示し、図2は、CPU温度とTimer1、CPU温度情報レジスタの動作を示している。図3は、STOPCLKリクエスト発生比率レジスタへの設定値とTimer2から生成される、STOPCLKリクエスト信号のTiming波形を示すものである。

【0021】パソコンの電源オンとともに図1のタイマー2(9)、STOPCLKリクエストデューティ比設定レジスタ5は、初期設定値により動作する。タイマー1(8)は一定周期で動作するタイマーで、図2のTimer1のTiming波形に示すように、そのタイマー周期でトリガが発生する。CPU1の近辺に実装されたCPU温度センサー2は、リアルタイムにCPUの温度情報をシリアル/パラレル変換3に送り続ける。

【0022】シリアル/パラレル変換3は、そのシリアルのCPU温度情報を常時パラレルデータに変換、Timer1のトリガのTimingでCPU温度情報レジスタ4へ設定値の更新が行われる。STOPCLKリクエストデューティ比5には、CPU温度におけるSTOPCLKリクエストのTiming波形の要素となる、CPUの温度毎のデューティ比の設定値が初期値で設定されている。もし、その初期値を変更する場合は、CPUアドレス/データバス101を介してCPU1により設定値を変更することや外付けROM8から設定値をダウンロードすることも可能である。

【0023】STOPCLKリクエストデューティ比設定レジスタ5には図3に示すSTOPCLKリクエスト波形の要素が設定されている。図3の説明を行う。ST

OPCLKリクエストデューティ比設定レジスタ5には、CPUの温度毎のSTOPCLKリクエストTimingのL:Hの割合のデューティ比が設定され、Lの割合を設定する。Lの割合を大きく設定（最大100%）する程CPUの性能は低下し、CPU温度が下がる傾向となる。Lの割合を小さくする（最大0%）とCPU性能は向上し、CPU温度も向上する傾向となる。

【0024】CPU温度情報レジスタ4からのCPU温度情報により、その内の1つの設定値が選択STOPCLKリクエストデューティ比データ106として選択されSTOPCLKリクエストTiming生成回路6へSTOPCLKリクエスト信号のソースとして送られる。STOPCLKリクエストTiming生成回路6は、それをもとにSTOPCLKリクエストのL:HのデューティTimingを生成する。

【0025】また、STOPCLKリクエスト信号の1周期のTimingはTimer2の周期によって決定される。Timer2の周期は、CPUアドレス/データバス101を介してCPU1により周期を変更することが可能で、アプリケーションとの組み合わせ動作によっては、この周期を調整することで、より効率的にCPU性能/温度比を制御できる。STOPCLKリクエストデューティ比設定レジスタ6では、STOPCLKリクエストセクタ7からの選択STOPCLKリクエストデューティ比データ106とタイマー2（9）からの信号によりCPUへのSTOPCLKリクエスト信号107を生成する。

【0026】

【実施例】CPU冷却が十分にできないノートパソコン等では、CPUの温度制御を行わず、そのまま温度が上昇すると、CPUの温度規格をオーバーし、CPUの異常動作によるパソコンの暴走、あるいはCPU自身の故障等の問題が発生する。本発明では、CPUの温度が上昇すると、CPUへSTOPCLKリクエスト信号を発生して、CPUの性能を低下させることによりCPUの温度の上昇を防ぐ機能を持っている。

【0027】本発明の実施例を図1～図5を用いて説明する。ここでは、例としてCPUの温度が65℃→80℃→75℃と変化した場合の動作を説明する。

【0028】CPU1は、電源オンとともに温度が上昇を始める。CPU1の近辺にCPU温度センサー2を実装し、CPUの温度を常時監視し続ける。CPU温度センサー2は、CPU温度情報をシリアルCPU温度データ102として常時シリアル/パラレル変換3へ送り続け、シリアル/パラレル変換3は、シリアルCPU温度データ102をパラレルCPU温度データ103としてCPU温度情報レジスタ4へ送り続ける。

【0029】CPU温度情報レジスタ4は、常時送られてくるパラレルCPU温度データ103をタイマー1（8）の周期で取り込みを行い、CPU温度情報レジ

スタ4の設定値は、タイマー1（8）の周期毎に更新されることになる。

【0030】本動作を図2を用いて説明すると、パラレル変換後のCPU温度103が、まず65℃であるので65℃の103のHEXコード8hが、タイマー1

（8）のトリガAの立ち上がりTimingでCPU温度情報レジスタ4へ取り込まれる〔CPU温度情報レジスタ4へのパラレルCPU温度データ103のHEXコードは、図4を参照。〕。

【0031】ここでは、例として25℃以下は、0h、65℃は8h、80℃はBh、75℃はAh、100℃以上はFh（その他は、省略〔図4参照〕）とした。次のトリガBのTiming時、パラレルCPU温度データ103は80℃であるため、BhのパラレルCPU温度データ103がCPU温度情報レジスタ4に、トリガCのTiming時には75℃であるのでAhのパラレルCPU温度情報データ103がCPU温度情報レジスタ4に更新される。

【0032】本例（図4参照）では、16種類（HEXコード0～Fh）のCPU温度が識別可能であるため、CPU温度情報レジスタ4では、HEXコードのデコードを行い、STOPCLKリクエストセクタ7にCPU温度デコード信号104として送る。図1のSTOPCLKリクエストデューティ比設定レジスタ5では、CPU温度毎に適切なSTOPCLKリクエストのL:Hのデューティ比が初期値として設定されている。初期値を変更する場合は、CPU1のCPUアドレス/データバス101を介してI/Oアクセスを行うことによって、CPU温度単位の変更が可能であり、また、外付けROM10に設定を組み込んでおき、電源オン時に、STOPCLKリクエストデューティ比設定レジスタ5へダウンロードするという機能もある。

【0033】図4では、例としてCPU温度に対するCPU温度情報レジスタ4のHEXコード及びCPU温度毎のSTOPCLKリクエストデューティ比設定レジスタ設定値の一例を記載した。本例では、CPU温度を25℃～100℃を5℃単位で区切り16個のブロックに分け、その16ブロック個々にSTOPCLKリクエストデューティ比設定レジスタ5の設定が可能である。ここでは、55℃まで0%、60℃で15%、65℃で20%、70℃で30%、～100℃まで徐々にSTOPCLKリクエストの発生比率を大きくしている。CPU温度が高いほど、STOPCLKリクエストの比率を大きくしてCPUの性能を低下させ、CPU温度を下げるというものである。

【0034】そのCPU温度毎の16種のSTOPCLKリクエストデューティ比データ105は、STOPCLKリクエストセクタ7へ渡される。STOPCLKリクエストセクタ7ではCPU温度情報レジスタ4からのCPU温度デコード信号104により、その16種

の設定値の中から1つを、選択STOPCLKリクエストデューティ比データ106としてSTOPCLKリクエストTiming生成回路6へ渡す。

【0035】STOPCLKリクエストTiming生成回路6は、選択STOPCLKリクエストデューティ比データ106とタイマー2(9)からの一定周期のトリガにより、CPUへのSTOPCLKリクエスト信号107のTimingを生成する。STOPCLKリクエストTiming生成回路6では、図2のTimingに示すように選択STOPCLKリクエストデューティ比データ106のSTOPCLKリクエストのデューティL:H比率をタイマー2(9)からの立ち上がりトリガ単位で1周期とする。タイマー2(9)の周期は初期値で設定されているが、CPU1によっても周期を変更することが可能。しかし、Timer1(8)の周期の整数分の1に設定しなければならない。

【0036】図5は、CPU温度が65℃→80℃→75℃と変化した場合のSTOPCLKリクエスト信号107の動作を示したものである。タイマー2(9)のタイマー値はタイマー1(8)のタイマー値の2分の1に設定されている。最初、CPU温度は65℃であるためTimer1のトリガAのTimingで、65℃時の20%のSTOPCLKリクエスト信号107〔図4より〕がCPU1に入力される。それでもCPU温度は上昇し、トリガB地点では、80℃になる。80℃では50%のSTOPCLKリクエスト信号107となり、CPU温度は低下し始める。トリガCの時点では、75℃となっており、40%のSTOPCLKリクエスト信号107が発生する。

【0037】上記、説明したように、本発明では、CPU温度単位に予め設定された、CPUへのSTOPCLKリクエストを発生する回路を持つことにより、ソフトウェアによるCPUの温度制御を必要としない。また、各CPU温度に応じた、最適な性能をSTOPCLKリクエストにより制御でき、CPU温度状況においての最適なCPU性能を発揮することが可能である。

【0038】

【発明の効果】以上、説明したように本発明は、ソフト

ウェアの制御を必要とせずに、CPU温度に応じて、STOPCLKリクエストの発生比率を制御することにより、その環境下での性能を十分に引き出すことが可能になる。また、ソフトウェアのプログラムミス等によって、ソフトウェアでの制御が不能な事態になっても、CPUの温度制御は行われるため、CPU温度が規格値を越えてしまい、故障するといった問題は発生しない。

【図面の簡単な説明】

【図1】本発明を示すブロック図である。

【図2】CPU温度サンプルタイミング図である。

【図3】STOPCLKリクエスト発生比率タイミング図である。

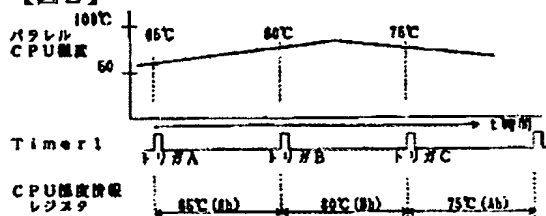
【図4】STOPCLKリクエスト発生比率設定の一実施例を示す図である。

【図5】本発明のSTOPCLKリクエストタイミング動作を示す一例の図である。

【符号の説明】

- 1 CPU
- 2 CPU温度センサー
- 3 シリアル/パラレル変換回路
- 4 CPU温度情報レジスタ
- 5 STOPCLKリクエストデューティ比設定レジスタ
- 6 STOPCLKリクエストタイミング生成回路
- 7 STOPCLKリクエストセレクト
- 8 タイマー1
- 9 タイマー2
- 10 ROM
- 101 CPUアドレス/データバス
- 102 シリアルCPU温度データ
- 103 パラレルCPU温度データ
- 104 CPU温度デコード信号
- 105 STOPCLKリクエストデューティ比データ
- 106 選択STOPCLKリクエストデューティ比データ
- 107 STOPCLKリクエスト信号
- 108 ROMデータ

【図2】



【図3】

